In the Claims:

Please replace claim 10 and add new claim 27, all as shown below.

1 - 3. (Canceled)

4. (Previously Presented): The computer-readable medium of claim 26 wherein the superclass includes logic to handle server side tasks.

5. (Previously Presented): The computer-readable medium of claim 26 wherein the wrapper class is generated in bytecode.

6. (Previously Presented): The computer-readable medium of claim 5 wherein bytecode is generated for vendor methods not implemented in the superclass.

7 - 8. (Canceled)

9. (Previously Presented): The computer readable medium of claim 26, wherein the application server supports Java Enterprise Edition.

10. (Currently Amended): A computer-readable medium carrying instructions for processing an invocation at a dynamically generated wrapper, comprising the steps of:

receiving, from an application, an invocation by a wrapper object, the wrapper object instantiated from a wrapper class, the wrapper class extended from a superclass which implements

Java Database Connectivity, Java Message Service and Java Connector Architecture, the ~~an~~ invocation directed to a wrapped resource adapter;

initiating pre-processing by calling a pre-invocation handler configured to execute server-side code;

calling the wrapped object;

receiving a result from the wrapped object;

initiating post-processing by calling a post-invocation handler configured to execute post processing server-side tasks; and

providing the result to the application, thereby enabling the application to access vendor specific extension methods of the wrapped resource adapter.

11 – 12. (Canceled)

13. (Previously Presented): The computer-readable medium of claim 10 wherein the server-side code executed by the pre-invocation handler includes transaction processing code.

14 – 15. (Canceled)

16. (Previously Presented): The computer-readable medium of claim 10 wherein the post-processing server-side tasks include transaction management.

17 – 23. (Canceled)

24. (Previously Presented): A computer-readable medium carrying instructions for processing an invocation at a dynamically generated wrapper, comprising the steps of:

receiving, from an application, a method invocation to a resource adapter;

calling a wrapper object for processing the method invocation wherein the wrapper object is dynamically generated from a resource adapter class;

initiating pre-processing by the wrapper object, wherein the wrapper object calls a pre-invocation handler configured to perform server side logic;

forwarding the method invocation to the resource adapter by the wrapper object on behalf of the application;

receiving a result of the method invocation from the resource adapter by the wrapper object;

initiating post-processing by the wrapper object, wherein the wrapper object calls a post-invocation handler configured to perform server-side logic; and

providing the result to the application, thereby enabling the application to access vendor specific extension methods of the resource adapter.

25. (Previously Presented): The computer-readable medium of claim 24 wherein the server-side logic includes at least one of transaction management, pooling, caching, tracing and profiling.

26. (Previously Presented): A computer-readable medium carrying instructions for dynamically generating a wrapper object, comprising the steps of:

receiving a resource adapter class at an application server;

performing reflection on the resource adapter class to identify interfaces implemented by the resource adapter class;

dynamically generating a wrapper class at runtime that extends from a superclass, wherein the superclass implements Java Database Connectivity, Java Messaging Service, or Java Connector Architecture interfaces, and the wrapper class implements the interfaces identified through reflection;

instantiating a wrapper object from the wrapper class; and

providing the wrapper object to an application that requires support for the interfaces implemented by the resource adapter class.

27. (New): The computer-readable medium of claim 26 wherein the superclass is statically predefined.